# Performance comparison of adapted Delaunay triangulation method over Nurbs for surface optimization problems

Bhattarai, Suyesh; Vichare, Parag; Dahal, Keshav

*Published in:*
ESM '2017

Published: 25/11/2017

*Document Version*
Peer reviewed version

# PERFORMANCE COMPARISON OF ADAPTED DELAUNAY TRIANGULATION METHOD OVER NURBS FOR SURFACE OPTIMIZATION PROBLEMS

Suyesh Bhattarai*
Parag Vichare
Keshav Dahal
Artificial Intelligence, Visual Communication and Networks Research Centre
University of the West of Scotland
Paisley, UK
Suyesh.bhattarai@uws.ac.uk

**KEYWORDS**
Delaunay Triangulation, NURBS, surface, optimization.

## ABSTRACT

Traditionally NURBS (Non-Uniform Rational Basis Spline) are used as the basis for defining free-form surfaces as they can define non-regular surfaces with minimal control points. However, they require parameters such as knot vectors and weights to configure a surface. Similarly, DT (Delaunay Triangulation) is proven and used widely for meshing, rendering and surface reconstruction applications, but its capability in freeform surface design for optimization is untested. Thus, this paper proposes Adapted Delaunay Triangulation (ADT) method which can generate a surface from scattered data points without any parameters. The paper presents a comparison of the performance of ADT method and NURBS fitting method for surface generation from scattered 3D coordinate points. This method was suggested so that the generated surface could be used in Stochastic Optimization Algorithm (SOA) methods and computational fluid dynamics applications (CFD) simultaneously. Data points that other 3D point clouds fitting methods would ignore as outliers are included in ADT method. Small change in each data point during optimization cycle should show a distinctive change in its output as SOA approaches depend on such differences for its optimal performance. Special consideration has been made for fast processing and rendering of the surface with minimum complexity (removing parameters such as knots and weights) and storage requirements as SOA methods demand generation of numerous surfaces to solve any problem.

## INTRODUCTION

Design optimization applications rely heavily on rendering surfaces and use various techniques for generation of these surfaces. Regular 2D planer facets can be created with straight or curved lines and the whole geometry for computer aided engineering (CAE) applications is created by merging these facets. These facets are usually built from well-defined base points. Generating surfaces from scattered points adds more complexity as undefined nature of the geometry may result in undesired, self-intersecting facets. Existing methods fit these points into Bezier or B-Spline surface, generating free-form surfaces (Narvaez, Narvaez, & Branch, 2010; Pizo & Motta, 2009). Usually, this method leaves out a number of points for configuring C1 or higher continuity surface. By contrast, other reported method by Boissonnat (J. D. Boissonnat, 1984) incorporates scattered point data in order to generate 3D elements for meshing solid geometry such as convex hull. Boissonnat and Cazals (Boissonnat & Cazals, 2002) and Amenta et. al. (Amenta, Bern, & Kamvysselis, 1998) reconstructed existing 3D surfaces from given sets of points, but with the assumption that i) the reference surfaces are smooth; ii) resulting surface will not have any open boundaries (such as solid models) and iii) normals to the surfaces are known.

Geometric design optimization for CAE application requires large scattered point cloud where every point has unique significance thus, cannot be filtered out for configuring geometry. Such examples include developing a freeform surface that could result in any shape as an optimization output. Most structural optimization methods study the strain and stress profile on the existing geometry and evaluate the most optimal design from the strain/stress graph (Madsen, Shyy, & Haftka, 2000; Papadrakakis, Lagaros, & Tsompanakis, 1998) but, it neglects the possibility of having an entirely new design unrelated to the existing one as discovered in the study by Linden (Linden, 2002). Especially for fluid dynamics studies, where a change in the interacting surface changes the overall nature of the flow, each change in the scattered point cloud is of importance.

This paper studies the previous works conducted in this area in section 2, explains the proposed Delaunay based method in section 3, compares the method against the widely used NURBS method in section 4, discusses the advantages and applications of the proposed method in section 5 and provides the conclusion in section 6.

## SURFACE CONSTRUCTION APPROACHES

Most surface generation work has been concentrated in surface reconstruction from a given set of scattered data points. The data points are obtained from vision based laser scanning sensor and are used to reconstruct these surfaces for rendering, graphics and pattern recognition. Research on configuring surfaces from point cloud has been classified by Boissonnat and Cazals (Boissonnat & Cazals, 2002) as:

1. Local projections (J. D. Boissonnat, 1984; Levin, 2004) develop surface as a function defined in a local reference domain. The surface is considered a graph of the function and approximated by triangulating in a moving projection plane or using least square function approximation techniques. These methods are fast but provide stretched and discontinuous surfaces with non-uniform and very sparse datasets.
2. Sculpting methods (J.-D. Boissonnat, 1984; J. D. Boissonnat, 1984) are based on removal of non-boundary facets from spatial arrangement, such as the convex hull. This method has performed well when the sampling is dense but reconstructed surface may not pass through all the sample points and may have additional holes.
3. Implicit methods (Boissonnat & Cazals, 2002; Hong-Kai, Osher, & Fedkiw, 2001; Ohtake, Belyaev, & Seidel, 2003) estimate a tangent plane from the sample data and uses distance to the plane as distance function. The zero-set of this function is then sampled at grid points and the surface is generated from these points. These methods require uniform and dense sampling for practical uses.
4. Deformable models (Amenta et al., 1998; Gary Wang, Dong, & Aitchison, 2001; Hoppe, DeRose, Duchamp, McDonald, & Stuetzle, 1992; Leal, Leal, & Branch, 2010) form an initial shell to which deformations are applied to minimize a function of energy and get closer to surface. Its performance depends largely on the initial guess which should be sufficiently close to the actual surface. These methods converge to local minima and could be significantly different from the true surface.

Sculpting and Deformable models based methods have an underlying assumption that all the surfaces are smooth and do not contain noise (Boissonnat & Cazals, 2002). Their performance has been commendable for surfaces without sharp edges and ample point density. But, these methods may fail to be robust and may require prohibitively large amounts of time to generate output for scattered point (J. D. Boissonnat, 1984; Hoppe et al., 1992). Thus, we will compare the performance of the proposed method based on local projections and NURBS based on implicit methods for our research problem of generating a surface from scattered points, inclusive of all the points.

Unlike polygons, NURBS are resolution independent and provide smooth curves and excellent continuity with fewer control points. But there are other parameters that greatly affect the topology of NURBS such as weights, knots and the degree of the curve (Narvaez et al., 2010). All these values must be perfectly coordinated to achieve the desired topology. NURBS requires a grid of control points that form the individual curves that can be moulded together to form a surface. This topology cannot be extended but can be patched with another such surface. In order to generate a NURBS surface from a set of scattered points, we first align the points cloud into a rectangular mesh. This mesh acts like the grid for the provided data set. The NURBS surface is generated using these points as the control points. The weights of each grid point are fixed as one and the degree of the spline curve is fixed as three to reduce variable parameters. First, three knot-vectors are defined as zero and the last three as one with uniformly spaced values in the remaining knots at the centre to ensure that the curves pass through the start and end points.

## ADAPTED DELAUNAY TRIANGULATION (ADT) METHOD

As summarized in the previous section, while NURBS surfaces have got distinguished advantages, they demand considerable computing resources for preparing geometry for CAE and CFD applications. This provides a scope for developing a light weight geometry preparation method for engineering analysis applications. The method proposed in this paper is for the specific purpose of real time applications on mechanical design optimization problems. The method utilizes Delaunay triangulation algorithm to generate a surface as a patch of triangular surfaces with straight and sharp edges.

### Algorithm

1. Define limits for the 3D points cloud

$$x_{min} \leq x \leq x_{max};\; y_{min} \leq y \leq y_{max};\; z_{min} \leq z \leq z_{max} \quad (1)$$

2. Define the number of points desired

$$[n] = \{1,\ldots\ldots\ldots, n\} \quad (2)$$

3. Generate the 3D points cloud with n points.

$$f(x) = \text{random}(\{x: x_{min} \leq x \leq x_{max}\}) \quad (3)$$
$$S = \{f(x_i, y_i, z_i), \forall\, i \in n\} \quad (4)$$

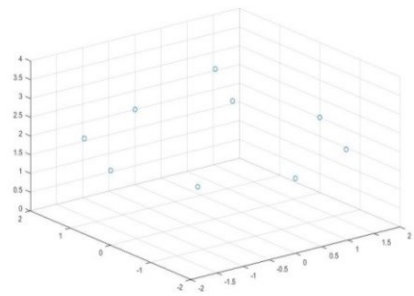


Fig.1. Generated points (Step 3)

4. Evaluate the spread of the coordinates by calculating their standard deviation

$$v(x) = \text{stdev}(\{f(x_i), \forall\, i \in n\}) \quad (5)$$
$$V = \{v(x), v(y), v(z)\} \quad (6)$$

5. Choose the coordinate axis with the minimum (or maximum) value of standard deviation to obtain depth axis of the surface. The chosen axis is the axis perpendicular to the generated surface

$$\text{floor.axis} := \text{axis with } \min\{V\} \tag{7}$$

6. If the values of standard deviation are equal follow the priority order of Z axis first and Y axis second.

$$\begin{aligned} \text{floor.axix} &:= \text{z-axis, if } v(x) = v(y) = v(z) \\ &:= \text{y-axis, if } \min\{V\} = v(x) = v(y) \end{aligned} \tag{8}$$

7. Create a set of 2D points with the remaining two coordinate axis values.

$$P = \{f(x_i, y_i, z_i),\ \forall\ i \in n\} - \{f(u_i)\} \tag{9}$$

where, u := z, if floor.axis is z axis
:= y, if floor.axis is y axis
:= x, if floor.axis is x axis

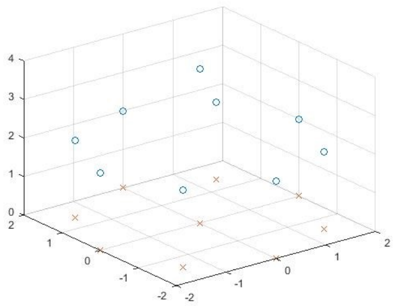


Fig.2. 2D projection (Step 7)

8. Apply 2D Delaunay algorithm to the generated set P.

9. Obtain the triangulation information (set of points that form a triangle) from 2D Delaunay output.

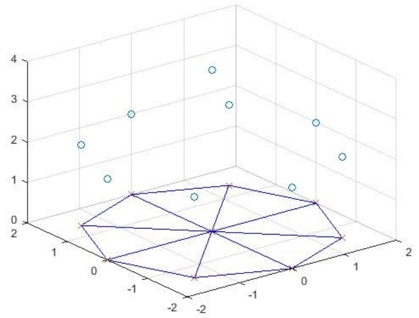


Fig.3. 2D Triangulation (Step 9)

10. Form a surface with the same triangulations in 3D space with respective x, y and z coordinates.

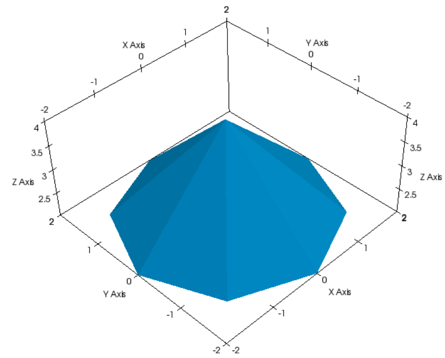
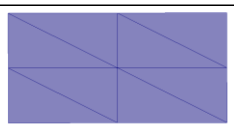
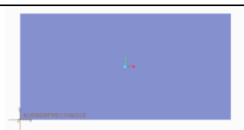
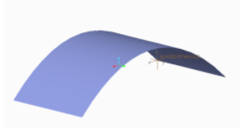
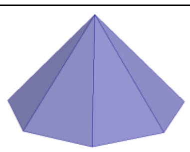
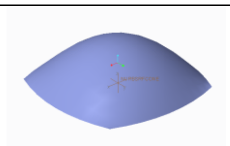
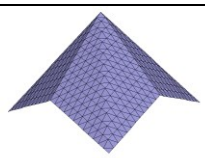
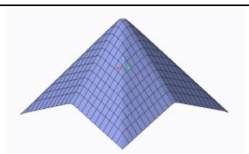
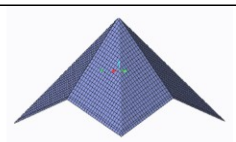



Fig.4. Surface Generation (Step 10)

This algorithm is basis of the method suggested in this paper.

Table 1: Comparison of ADT and NURBS

| ADT method | | NURBS method | |
|---|---|---|---|
|  | Number of points: 9<br>File size: 298 bytes<br>Time taken: 0.0035 sec |  | Number of points: 9<br>Number of control points: 3<br>Number of knots: 6<br>File size : 1,376 bytes<br>Time taken: 0.0118 sec |
|  | Number of points: 9<br>File size: 298 bytes<br>Time taken: 0.0026 sec |  | Number of points: 9<br>Number of control points: 3<br>Number of knots: 6<br>File size : 1,376 bytes<br>Time taken: 0.0123 sec |
|  | Number of points: 9<br>File size: 304 bytes<br>Time taken: 0.0045 sec |  | Number of points: 9<br>Number of control points: 3<br>Number of knots: 6<br>File size : 1,376 bytes<br>Time taken: 0.0156secs |
|  | Number of points: 625<br>File size : 25,963 bytes<br>Time taken: 0.0073 sec |  | Number of points: 625<br>Number of control points: 25<br>Number of knots: 28<br>File size : 33,128 bytes<br>Time taken: 0.0312secs |
|  | Number of points : 10,000<br>File size : 490,737 bytes<br>Time taken : 0.0847 sec |  | Number of points: 10,000<br>Number of control points: 100<br>Number of knots: 103<br>File size : 509,084 bytes<br>Time taken: 0.3814secs |

## EXPERIMENTATION

The outputs generated from the different input values with the ADT based method and the NURBS based method is compared in this section. The comparison in Table 1 shows that there is a distinct advantage of using the ADT method over the NURBS method for optimization applications with minimal processing of the random data fed as input.

## DISCUSSION

The advantages of the two methods employed in this paper can be briefed as following from the information collected from the above table. A typical case study with 10,000 points is considered for the comparison below.

1. Speed of generation: The most important factor while generating surfaces during optimization is the speed in which the geometry is created. The experiment shows that ADT method is 4.45 times faster than NURBS based method. This provides a massive advantage over the NURBS based method while generating multiple geometries.

2. Storage memory: The other important factor in optimization problems is the memory requirement and with the ADT method we get a 3.61% reduction in the total memory requirement for 10,000 points. And for geometry with 625 points, we get a 21.62% reduction in the memory requirement for ADT method. Such reduction in memory requirements enable running the simulation for even more geometries and allow more exhaustive search in SOA.

3. Geometric Continuity: The image generated from ADT method is made from joining together of flat triangular surfaces and hence provides a C0 continuity with respect to the adjacent surface. Whereas the NURBS method fits in the surface so that the continuity is maintained at C1 or above as specified by the codes. The ADT surface will look patched and pixelated while the smoothness of the NURBS surface adds to the aesthetic appeal for such surfaces.

4. Ability: The C0 continuity of ADT method allows for the geometry to incorporate sharp corners and a sudden change in the gradient of the surface topology, but NURBS being a fitting method does not allow for sharp corners and sudden change in the gradient of the surface. With designs requiring sharp edges and corners, two or more NURBS surfaces will have to be patched together. For designs requiring a smooth transition, ADT will require dense point cloud in such area of the geometry.

5. Pre-processing of Input data: ADT method takes the entire dataset as a whole and processes it all together to form the surface so pre-processing of the input data is not required. For the NURBS method, the data must be pre-processed and arranged in a grid to fit the basis spline curves. This pre-processing of data increases the complexity of this method.

6. Input data inclusion: The ADT method includes all the points on the surface and hence has no outliers. Every point lies on the surface of the geometry generated from ADT method. This ensures that a single change in the input data shows some drastic change in the output surface. Whereas in NURBS method, the surface is fitted based on predefined degree equation and hence some points do not lie on the surface of the geometry. This reduces the impact of changing a single point on the entire geometry. In SOA applications, it is desirable to have definitive changes in the geometry from a change in a coordinate point.

7. Variables: The ADT method only requires the coordinate values of the point cloud to generate a surface. But the NURBS method requires additional parameters such as weights, knots and degrees to generate the surface. These additional parameters may require some changes depending on the nature of the points cloud. In SOA applications, these parameters increase the complexity of the problem and may fail to provide a suitable surface as an output.

8. Robustness: The ability of Delaunay methods have been proven from studies carried out in the past. It is able to handle a large number of scattered points. These points need not be arranged in a grid, but the distribution must be fairly uniform to avoid holes and unwanted features. The NURBS method needs the points to be arranged in a proper grid and hence is less robust as it might be difficult to form grids from some groups of scattered points. The other parameters, such as weights and knot vectors, need their values to be well defined to achieve the desired NURBS surface. When dealing with numerous scattered point cloud sets, the same parameter values might not yield the best results.

9. Compatibility: The ADT method generates the surfaces in VTK format and this format can be used with any open source rendering and simulation packages. The NURBS format was developed for industrial use and is mostly associated with commercial software packages. It makes the ADT method easier to access for the general public.

10. Applications: This overall comparison shows that ADT method is ideal for use in SOA applications to determine the initial design of any surface whose performance can

be determined from CFD simulations. The NURBS method output is smooth and aesthetically pleasing and can hence be used in imaging and rendering applications. It can also be used to generate a geometry based on the final output from ADT method and run simulations on it.

## CONCLUSION

Current developments in graphics and surface rendering are demanding smoother surface finish and aesthetics for graphical interfaces. Such applications require considerable computational power at hand to process limited graphical information on the screen. Other applications require generating numerous geometries with constraints of time, computational power and storage capacity. The proposed ADT method is robust and provides about 4 times faster and simpler construction with 3-20% less memory requirement to generate surfaces that are compatible with multiple simulation packages and can be used together with SOA. The proposed method is dependent only on the coordinate points and hence provides consistent outputs for the same data while allowing sudden changes in the gradient and sharp corners that other freeform methods cannot. These qualities make this method very desirable for applications where the performance of the surface is dependent on its geometry, especially where a small change in one portion of the geometry may call for major changes in the remaining portion such as fluid flow over the surface. This method was suggested to be used together with computational fluid dynamics simulation software and stochastic optimization algorithms to produce an optimal surface for geometric design problems.

## ACKNOWLEDGEMENT

The first author likes to acknowledge the support provided by the EU Erasmus Mundus project SmartLink (552077-EM-1-2014-1-UK-ERA) to carry out this research at the University of the West of Scotland, UK.

## REFERENCES

Amenta, N., Bern, M., & Kamvysselis, M. (1998). A new Voronoi-based surface reconstruction algorithm. Paper presented at the Proceedings of the 25th annual conference on Computer graphics and interactive techniques.

Boissonnat, J.-D. (1984). Geometric structures for three-dimensional shape representation. ACM Trans. Graph., 3(4), 266-286. doi:10.1145/357346.357349

Boissonnat, J.-D., & Cazals, F. (2002). Smooth surface reconstruction via natural neighbour interpolation of distance functions. Computational Geometry, 22(1), 185-203. doi:http://dx.doi.org/10.1016/S0925-7721(01)00048-7

Boissonnat, J. D. (1984). Geometric Structures for Three-Dimensional Shape Representation. ACM Transactions on Graphics, 3(4), 21.

Gary Wang, G., Dong, Z., & Aitchison, P. (2001). Adaptive Response Surface Method - A Global Optimization Scheme For Approximation-Based Design Problems. Engineering Optimization, 33(6), 707-733. doi:10.1080/03052150108940940

Hong-Kai, Z., Osher, S., & Fedkiw, R. (2001). Fast surface reconstruction using the level set method. Paper presented at the Proceedings IEEE Workshop on Variational and Level Set Methods in Computer Vision.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., & Stuetzle, W. (1992). Surface reconstruction from unorganized points. SIGGRAPH Comput. Graph., 26(2), 71-78. doi:10.1145/142920.134011

Leal, N., Leal, E., & Branch, J. W. (2010). Simple Method for Constructing NURBS Surfaces from Unorganized Points. In S. Shontz (Ed.), Proceedings of the 19th International Meshing Roundtable (pp. 161-175). Berlin, Heidelberg: Springer Berlin Heidelberg.

Levin, D. (2004). Mesh-Independent Surface Interpolation. In G. Brunnett, B. Hamann, H. Müller, & L. Linsen (Eds.), Geometric Modeling for Scientific Visualization (pp. 37-49). Berlin, Heidelberg: Springer Berlin Heidelberg.

Linden, D. S. (2002). Antenna Design Using Genetic Algorithm. Paper presented at the GECCO.

Madsen, J. I., Shyy, W., & Haftka, R. T. (2000). Response surface techniques for diffuser shape optimization. AIAA Journal, 38(9), 1512-1518. doi:10.2514/2.1160

Narvaez, N. E. L., Narvaez, E. A. L., & Branch, J. W. (2010). Automatic Construction of NURBS Surfaces from Unorganized Points. DYNA, 78(166), 8.

Ohtake, Y., Belyaev, A., & Seidel, H.-P. (2003). A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. Paper presented at the Shape Modeling International.

Papadrakakis, M., Lagaros, N. D., & Tsompanakis, Y. (1998). Structural optimization using evolution strategies and neural networks. Computer Methods in Applied Mechanics and Engineering, 156(1), 309-333. doi:http://dx.doi.org/10.1016/S0045-7825(97)00215-6

Pizo, G. A. I., & Motta, J. M. S. T. (2009). 3D Surface Generation from Point Clouds Acquired from a Vision-based Laser Scanning Sensor. Paper presented at the 20th International Congress of Mechanical Engineering, Gramado, Brazil.